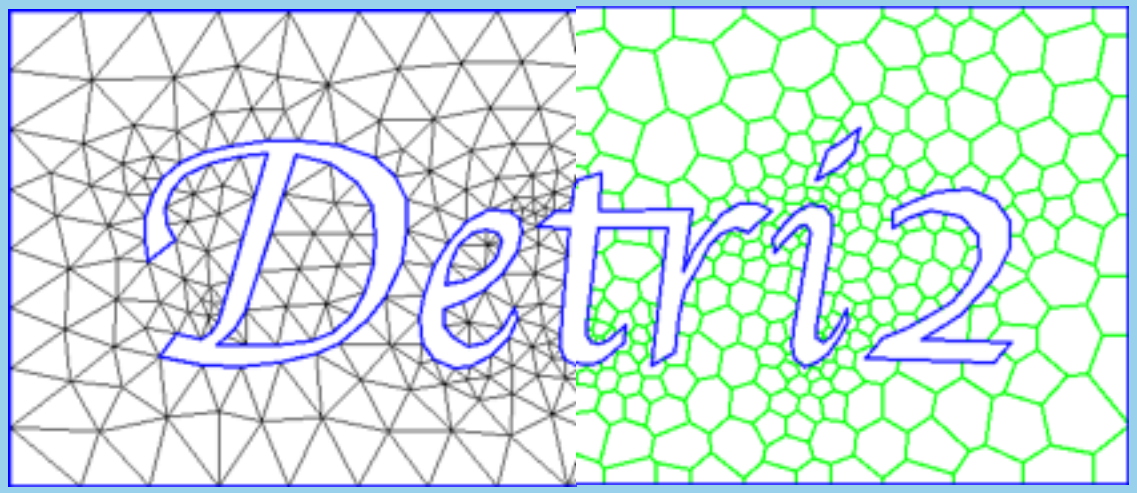


A 2d Constrained Delaunay-Voronoi Mesh Generator

Hang Si

Weierstrass Institute for Applied Analysis and Stochastics (WIAS) Berlin

2020-05-28

WIAS Berlin

Introduction

1. Detri2 is a C++ program to generate triangular meshes from arbitrary polygonal domains.
2. It generates (weighted) Delaunay triangulations for (weighted) point sets in 2d.
3. It uses constrained Delaunay algorithms to conform the input line segments and maintain the locally Delaunay property of the mesh topology.
4. It uses Delaunay refinement algorithms to generate meshes with good angle bounds.
5. It can generate adaptive meshes corresponding to input mesh sizing functions.

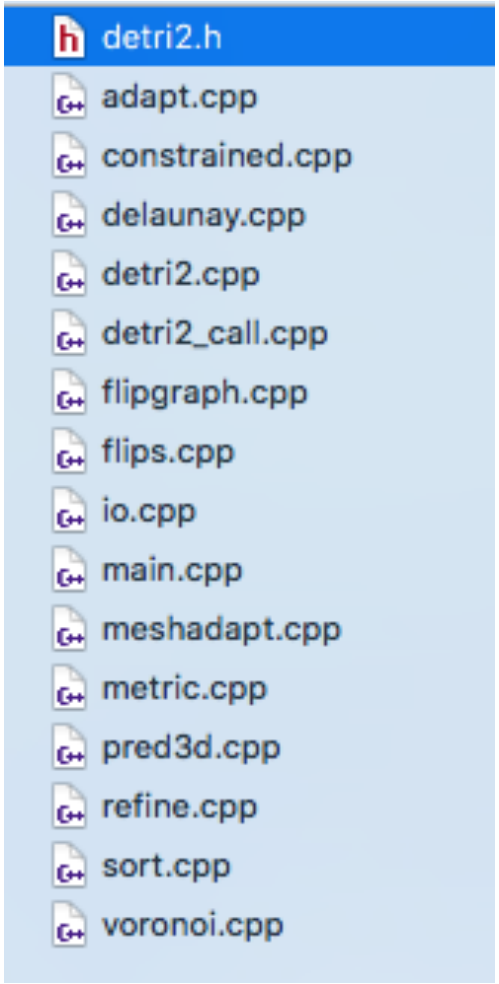
Why develop a new code?

1. Triangle (J. Shewchuk) is very good open source code for 2d Delaunay mesh generation. It is robust and efficient, and it is very well documented.
2. It is not suitable for educational purposes.
3. Adding new features into Triangle is difficult, e.g., (anisotropic) mesh adaptation, HDE, etc.
4. It does not support constrained weighted Delaunay triangulations (Math+ EF2-4).

```
/* Regular/weighted triangulations are incompatible with PSLGs */
/*   and meshing.                                               */
if (b->weighted && (b->poly || b->quality)) {
  b->weighted = 0;
  if (!b->quiet) {
    printf("Warning:  weighted triangulations (-w, -W) are incompatible\n");
    printf("  with PSLGs (-p) and meshing (-q, -a, -u).  Weights ignored.\n"
           );
  }
}
```

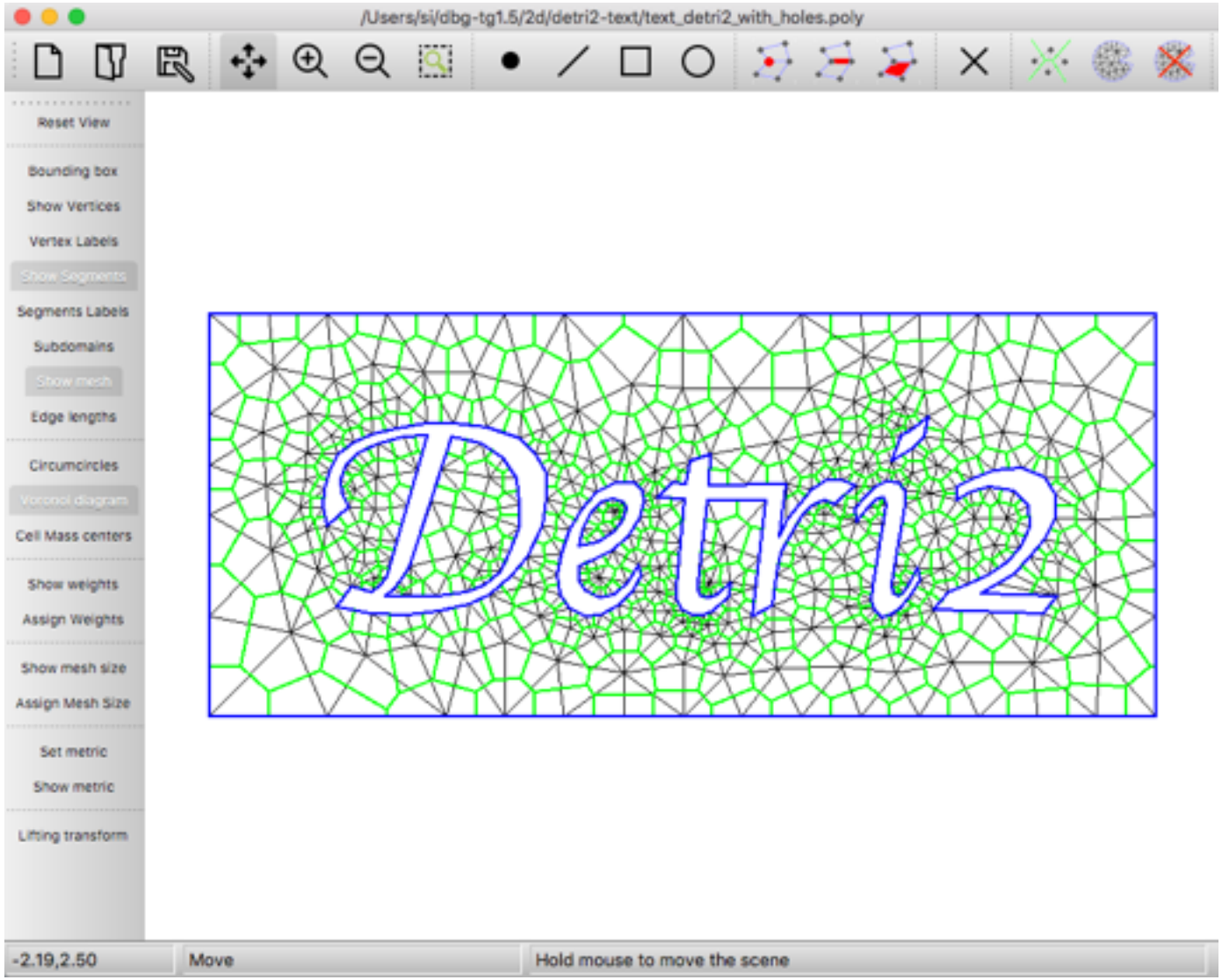
source code of Triangle

Detri2 source code



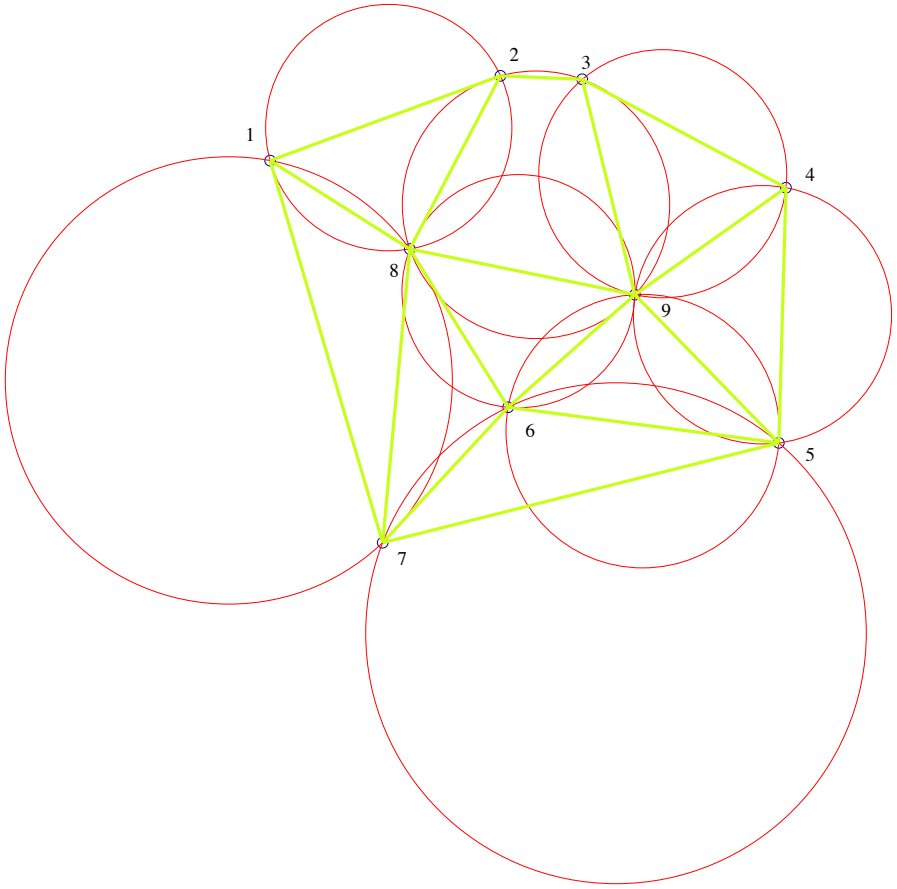
www.wias-berlin.de/people/si/detri2.html

detri2qt


/Users/si/dbg-tg1.5/2d/detri2-text/text_detri2_with_holes.poly
Reset View
Bounding box
Show Vertices
Vertex Labels
Show Segments
Segments Labels
Subdomains
Show mesh
Edge lengths
Circumcircles
Voronoi diagram
Cell Mass centers
Show weights
Assign Weights
Show mesh size
Assign Mesh Size
Set metric
Show metric
Lifting transform
Detri2
-2.19,2.50
Move
Hold mouse to move the scene

Delaunay Triangulations

Empty circumscribed circles



Delaunay triangulation

B.N. Delone (1890-1980)

Voronoi diagrams

$$V_p = \{\mathbf{x} \in \mathbb{R}^2 \,|\, \|\mathbf{x} - \mathbf{p}\| \leq \|\mathbf{x} - \mathbf{q}\|, \forall \mathbf{q} \in S\},$$

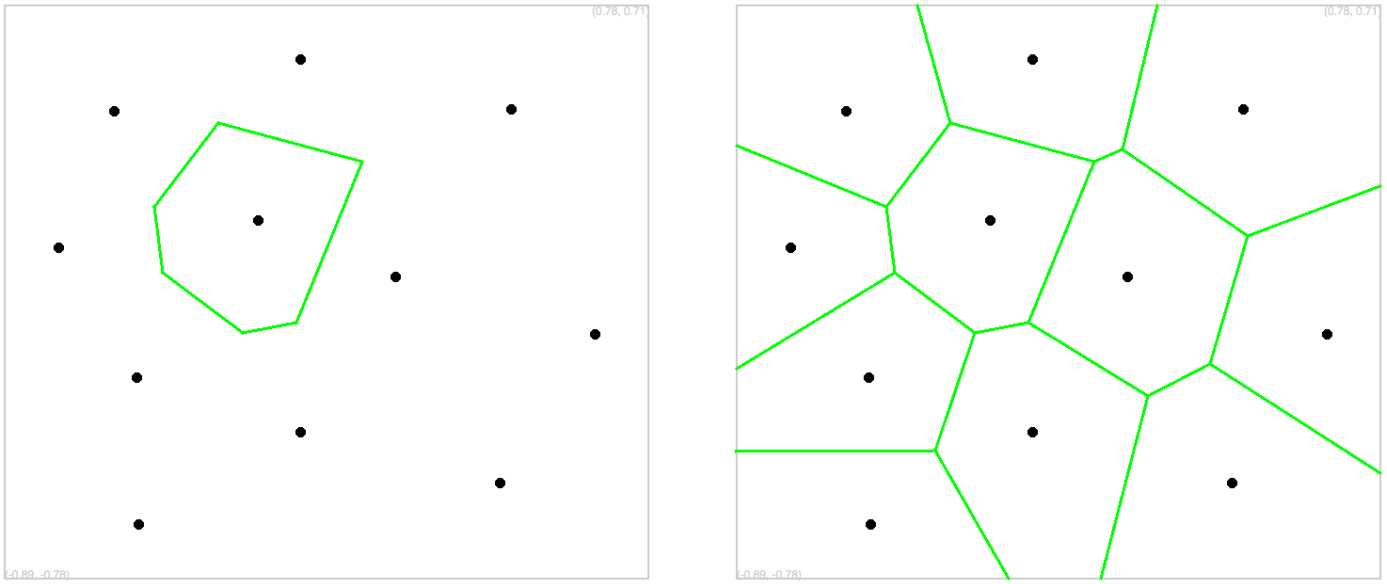


FIGURE 1. A Voronoi region (left) and the Voronoi diagram (right) of a two-dimensional point set.

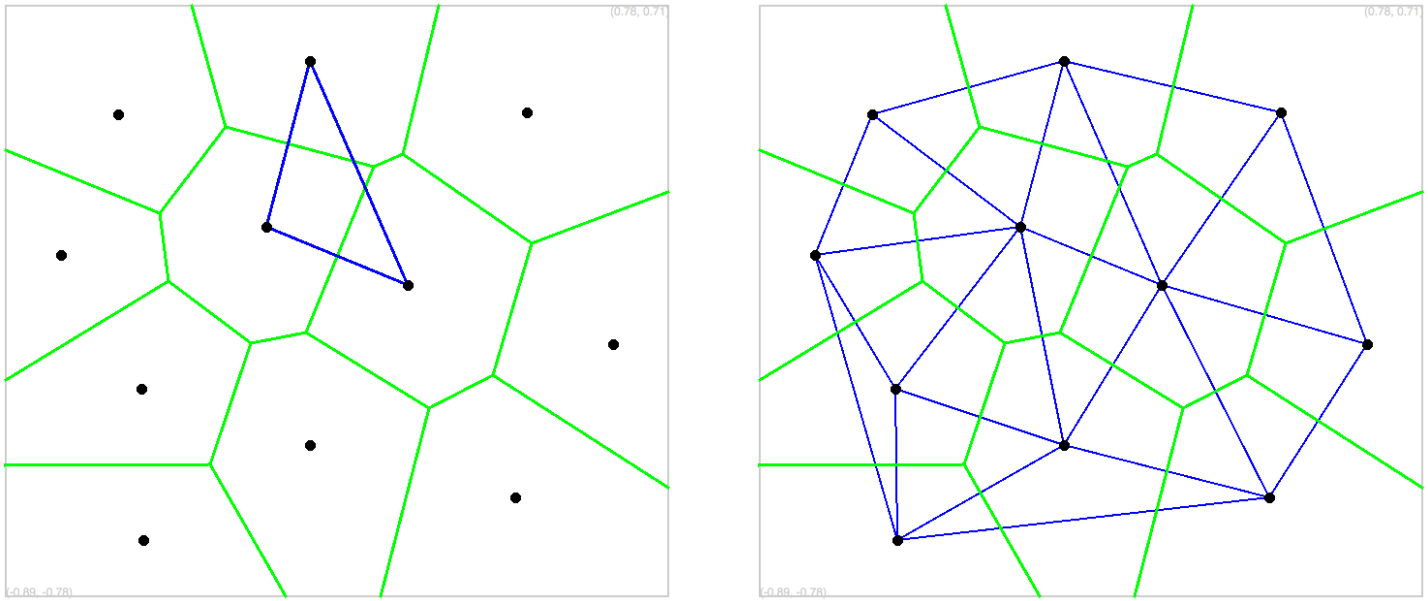
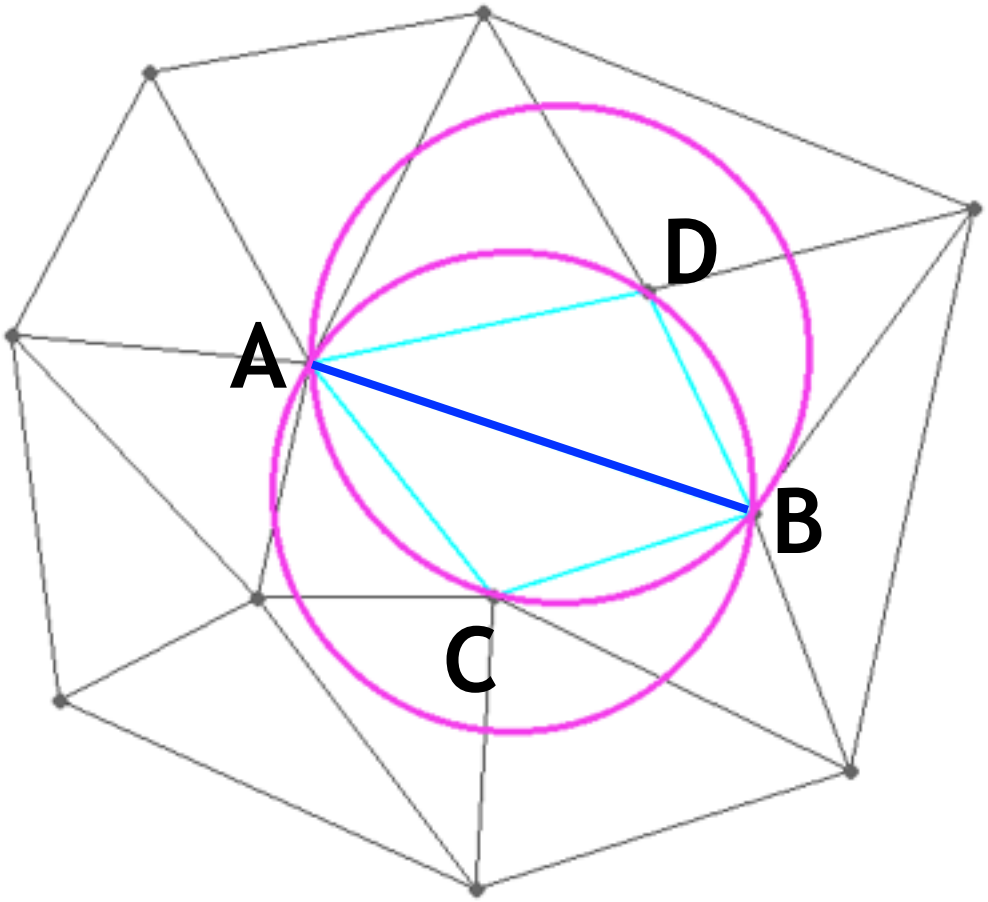


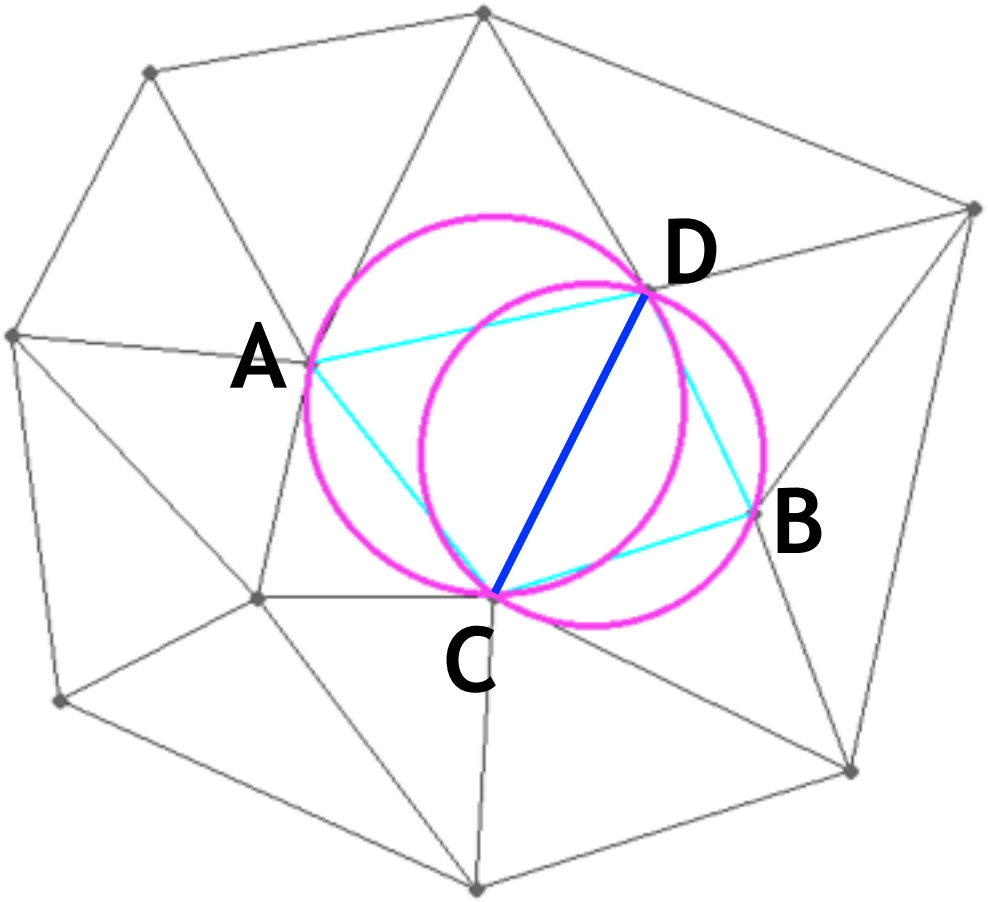
FIGURE 2. Three dual edges of the Voronoi edges (Left) and the dual diagram (Right). If no four or more points of the point set share a common circle, then this dual diagram is the Delaunay triangulation of this point set.

Lawson’s flip algorithm

Locally (weighted) Delaunay



AB is not locally Delaunay

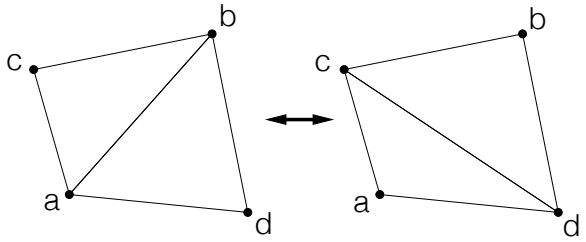


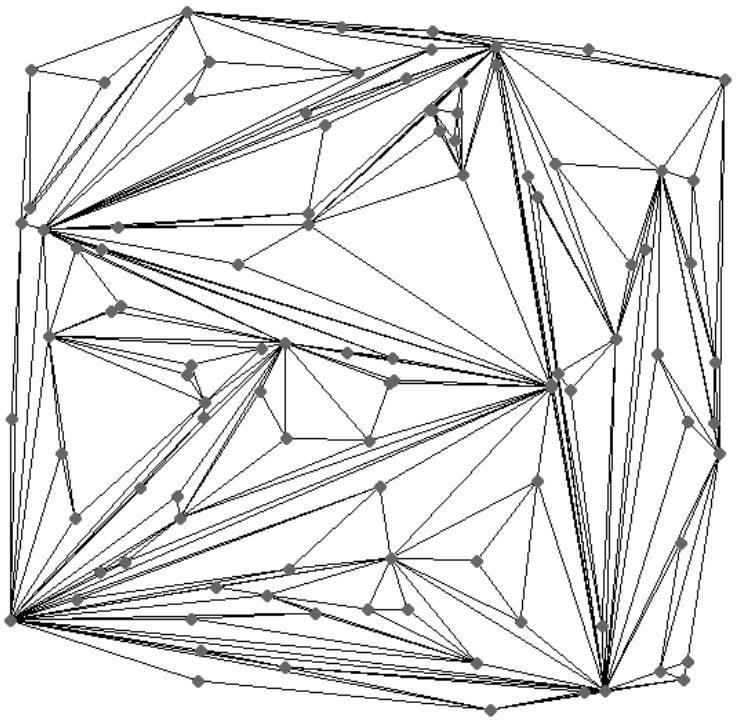
CD is locally Delaunay

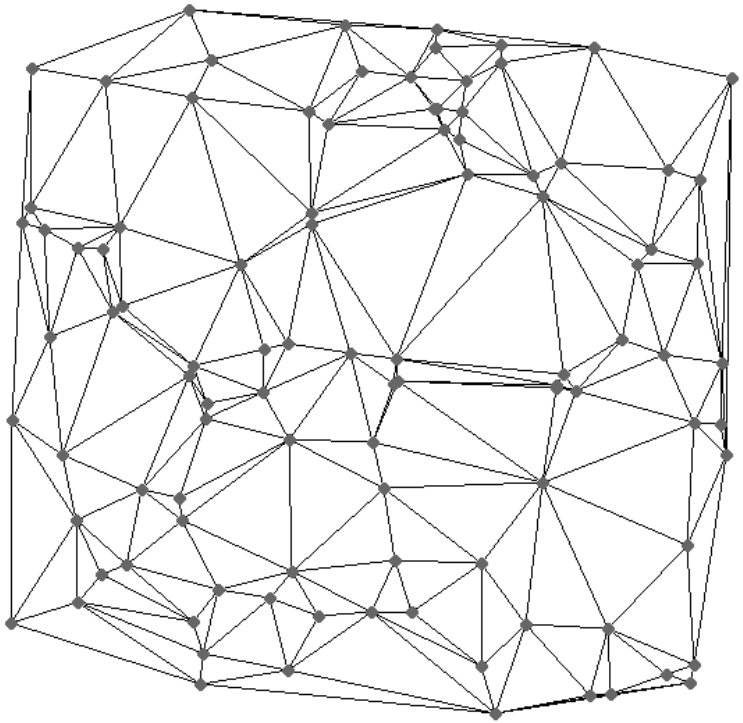
Lawson's Flip Algorithm [1977]

- Let $S = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ be a finite set of points in $\mathbb{R}^2$.
- Compute an initial triangulation $\mathcal{T}$ of a point set S.

while $\exists$ a locally non-Delaunay edge **ab** $\in \mathcal{T}$
    flip **ab**;
end while







A lower bound on the number of flips

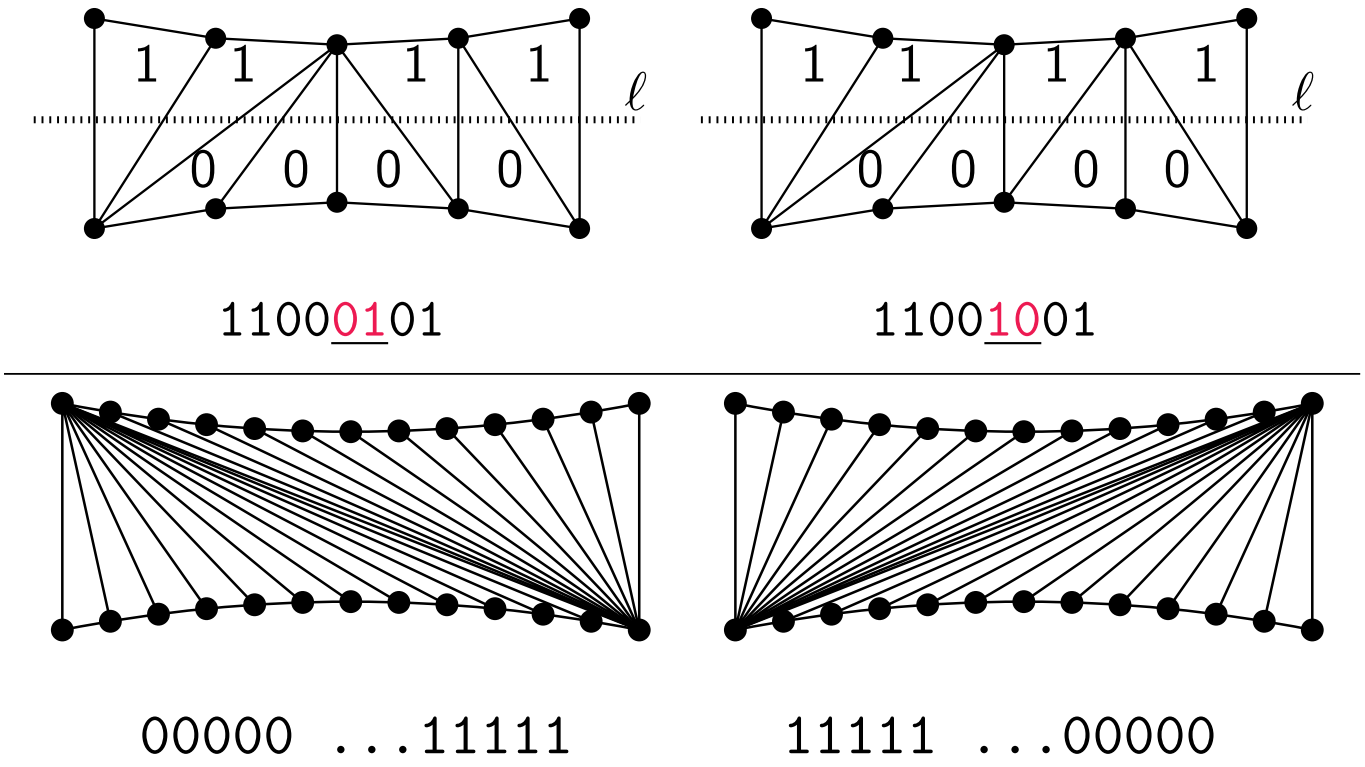
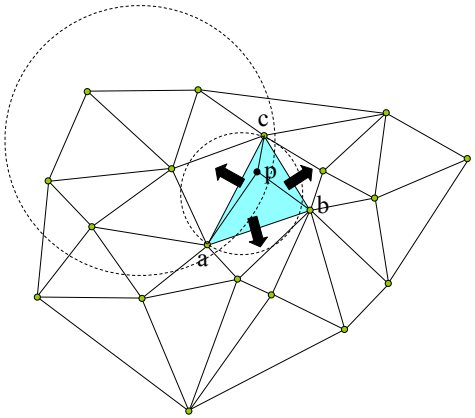


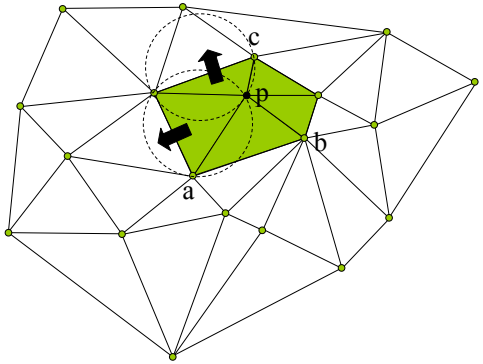
FIGURE 20. A lower bound example (Figures from A. Pilz (IST TU-Graz)).

Incremental Construction

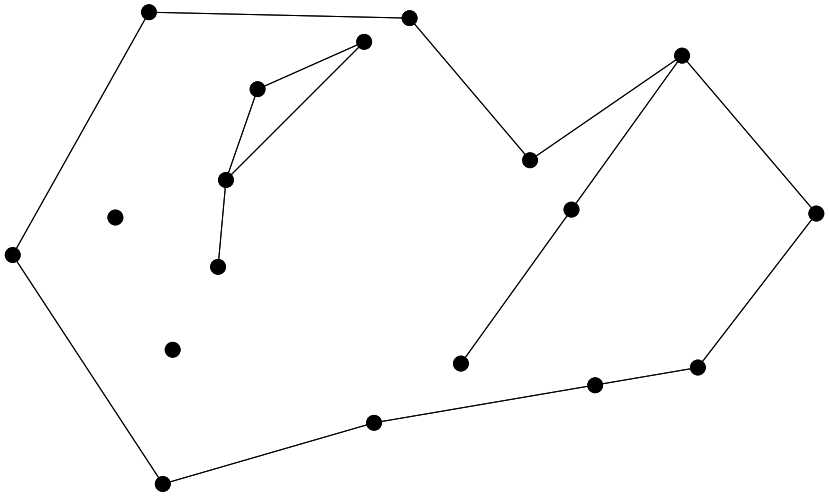
Algorithm: IncrementalFlip($S = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$)
Input: a sequence S of n points in $\mathbb{R}^2$;
Output: the Delaunay triangulation $\mathcal{D}$ of S;

1 initialize $\mathcal{D}_0$ with only one larger triangle $t_{\mathbf{xyz}}$;
2 **for** $i = 1$ to n **do**
3 find the triangle $\tau \in \mathcal{D}_{i-1}$ containing $\mathbf{p}_i$;
4 insert $\mathbf{p}_i$ by a 1-3 flip;
5 initial the stack L with link edges of $\mathbf{p}_i$;
6 LawsonFlip(L);
7 **endfor**
8 remove all triangles containing $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$ from $\mathcal{D}_n$;

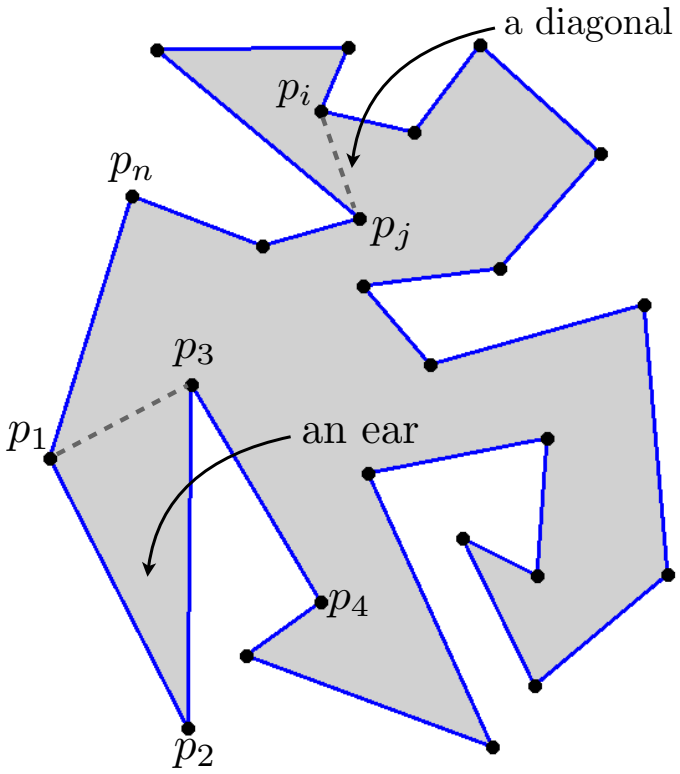


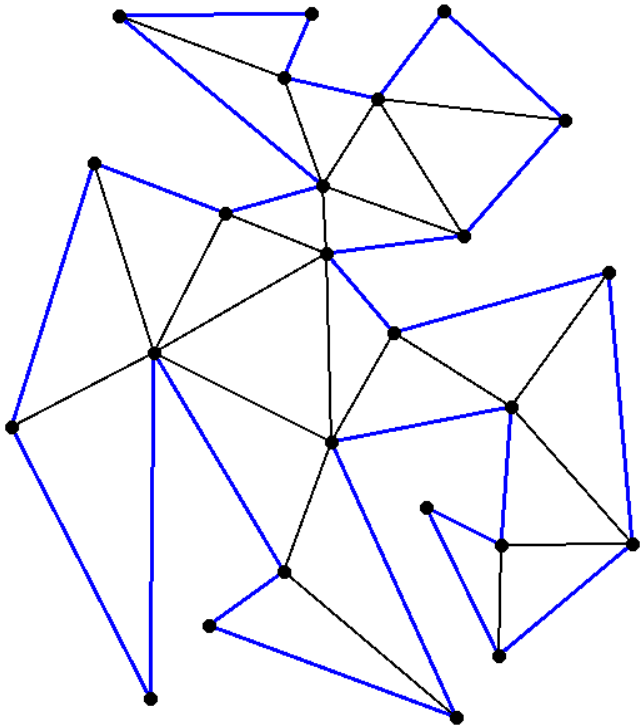


Meshing a Planar Straight Line Graph (PSLG) in plane



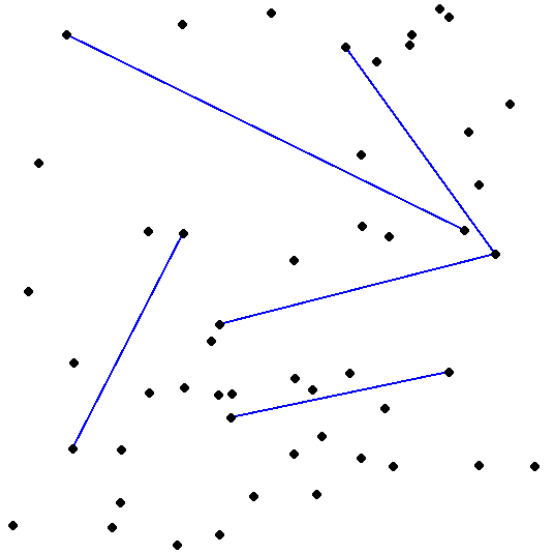
Triangulations of simple polygon



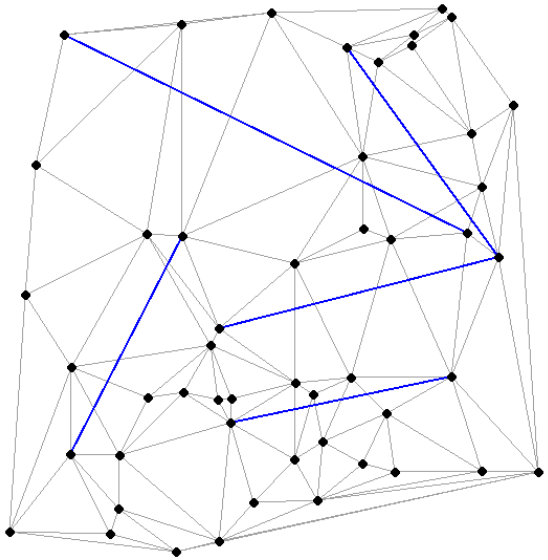


Theorem 1.1 ([8]). *Every simple polygon with more than* 3 *vertices has at least two ears.*

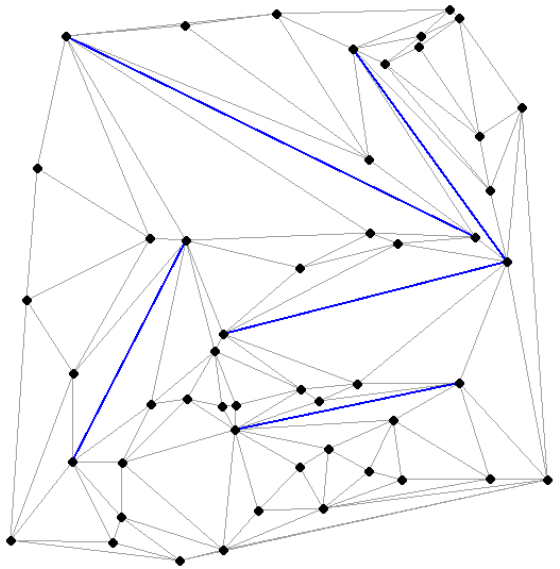
Constrained Triangulations of PSLGs



A PSLG (S, L)



The DT of S



A CT of (S, L)

Recover missing edges

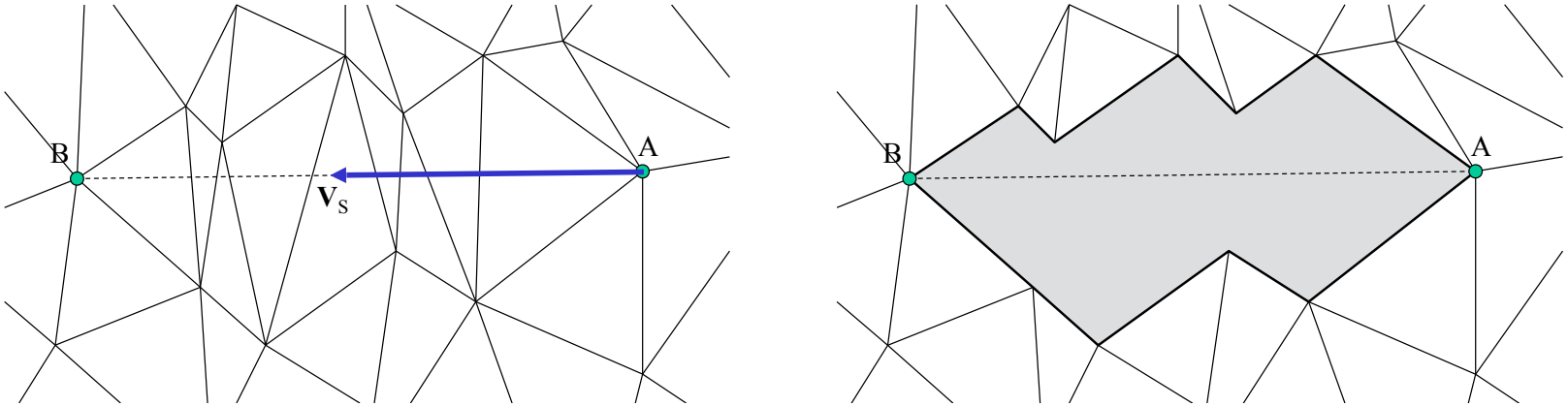
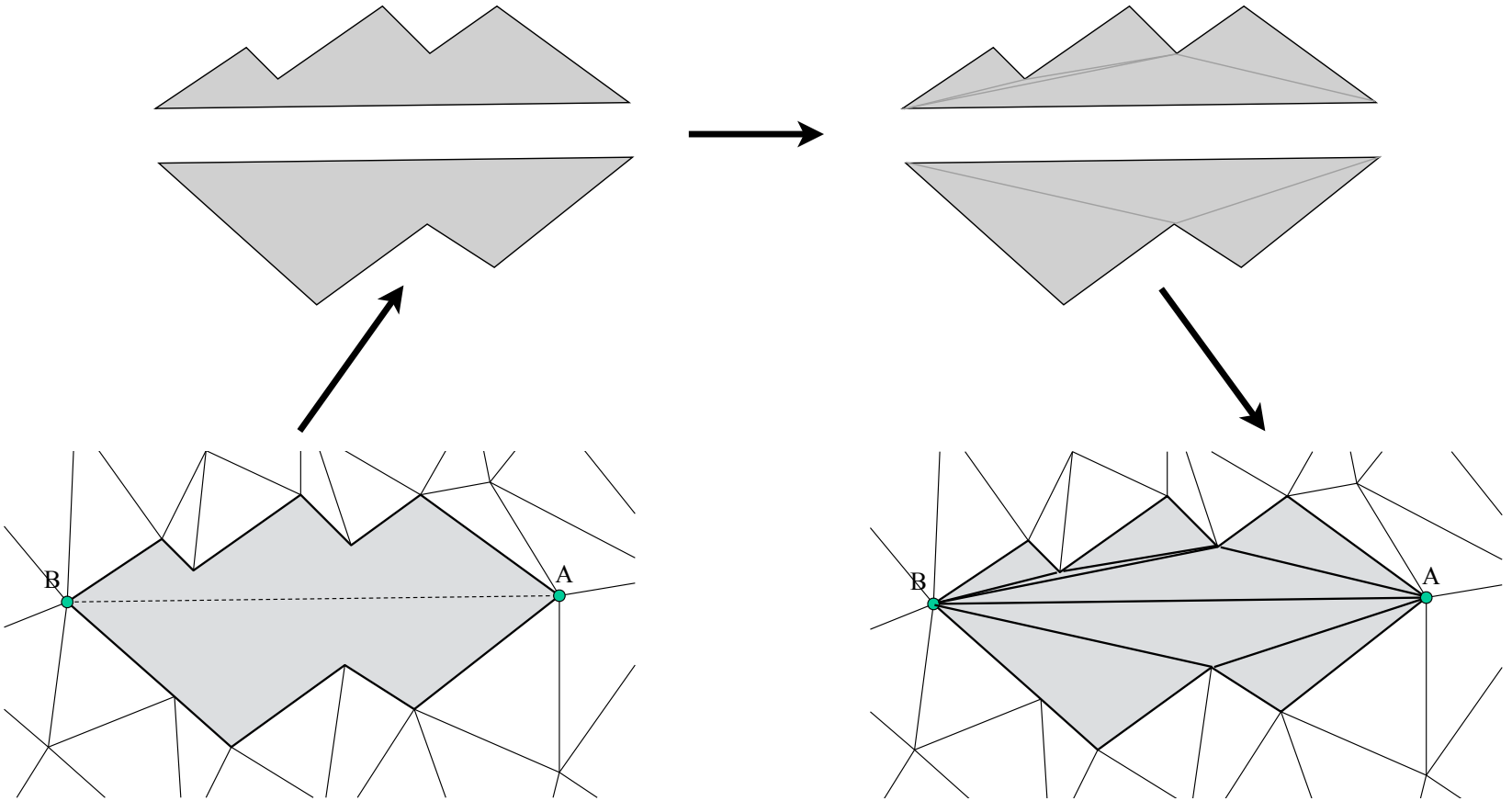
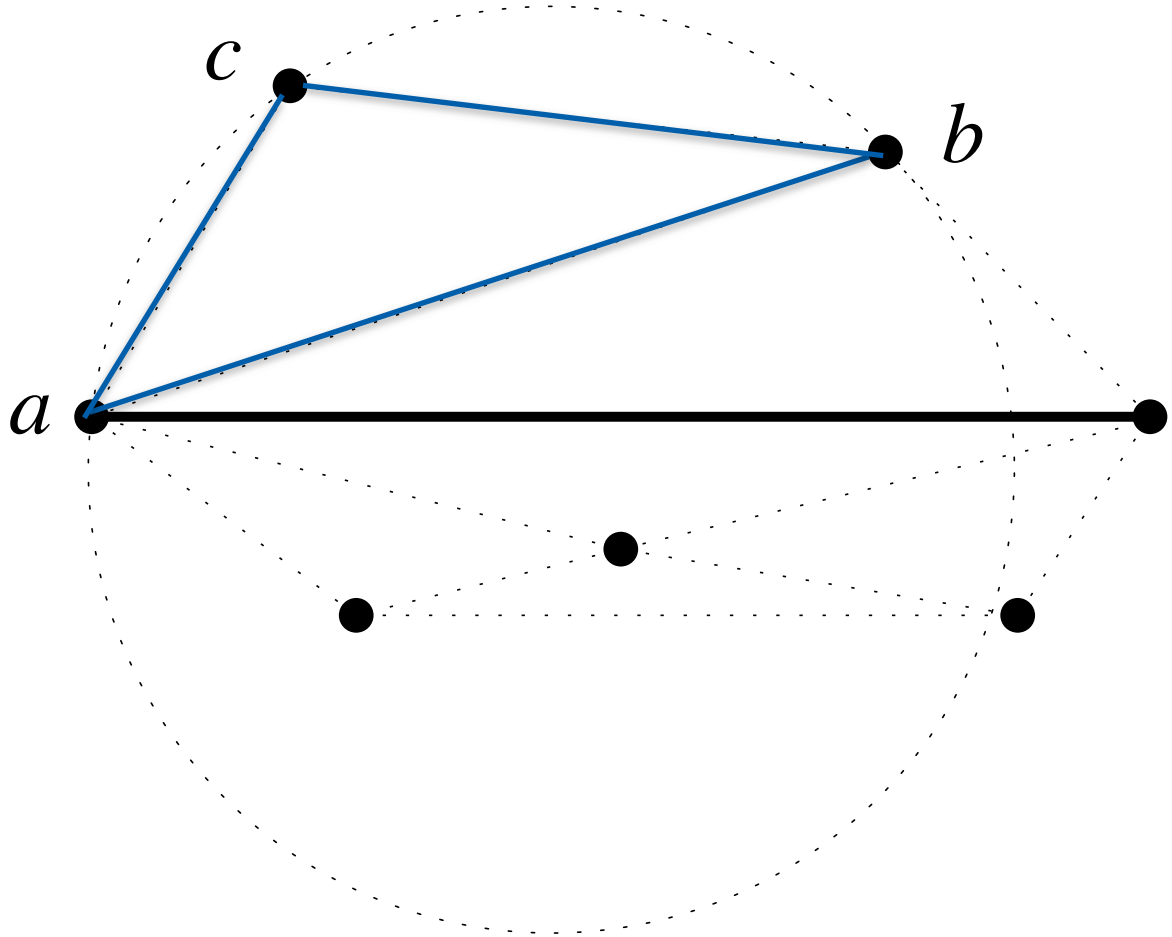


FIGURE 10. Left: search an edge e_{AB} in a constrained triangulation. Right: the cavity of the missing edge e_{AB}. (Figures from S. Owen).

Recover missing edges



Constrained Delaunay Triangulations



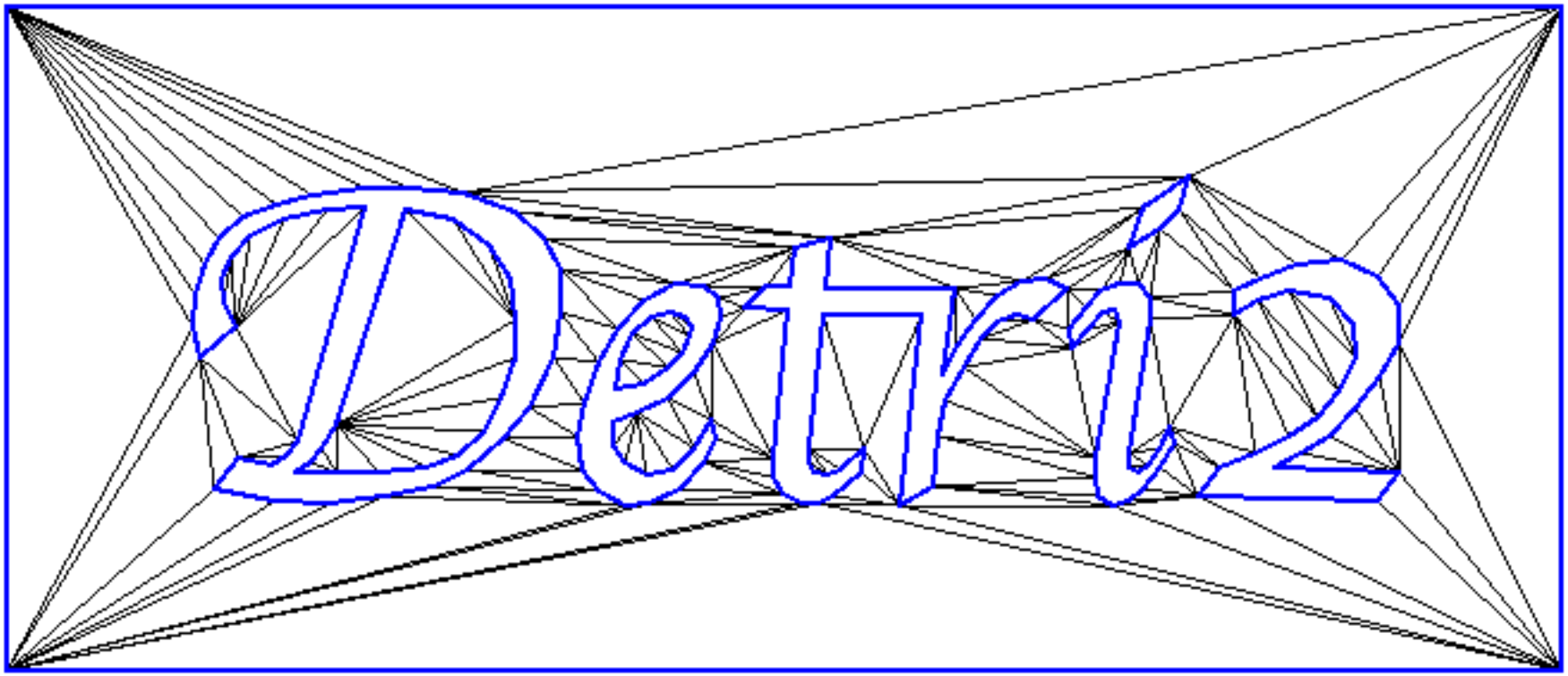
Incremental CDT

Algorithm: IncrementalCDT(S, L)
Input: A PSLG (S, L), $k := |L|$;
Output: the CDT $\mathcal{T}$ of (S, L);

```
1  construct an initial CDT T_0 of S;
2  for i = 1 to k do
3      if s_i ∈ L and s_i ∉ T_{i-1} then
4          RecoverEdge(s_i, T_{i-1});
5          Let E be the set of new edges in T_{i-1};
6          ConstrainedLawsonFlip(E);
7      endif
8  endfor
```

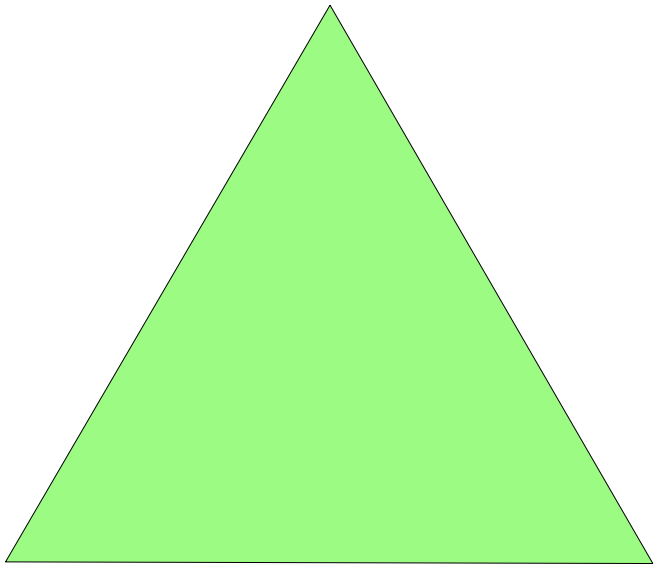
FIGURE 17. The incremental constrained Delaunay triangulation algorithm.

A constrained Delaunay triangulation

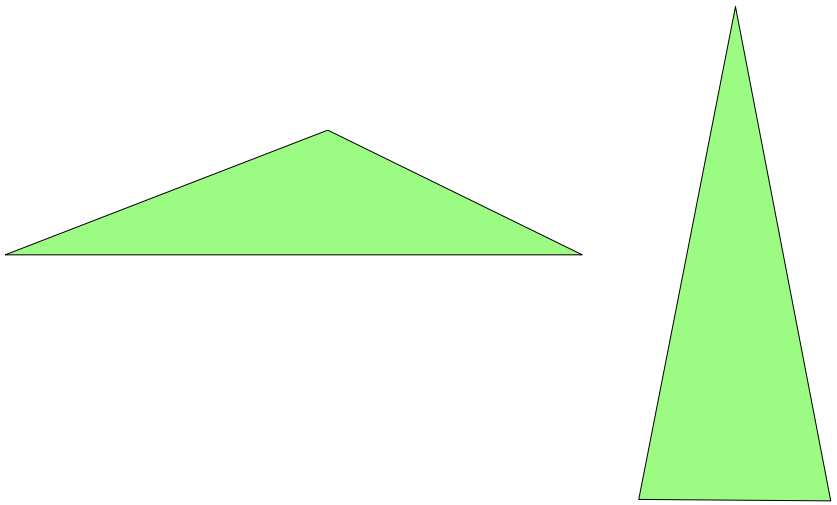


Quality Mesh Generation

Quality of triangles



‘Good’



‘Not Good’

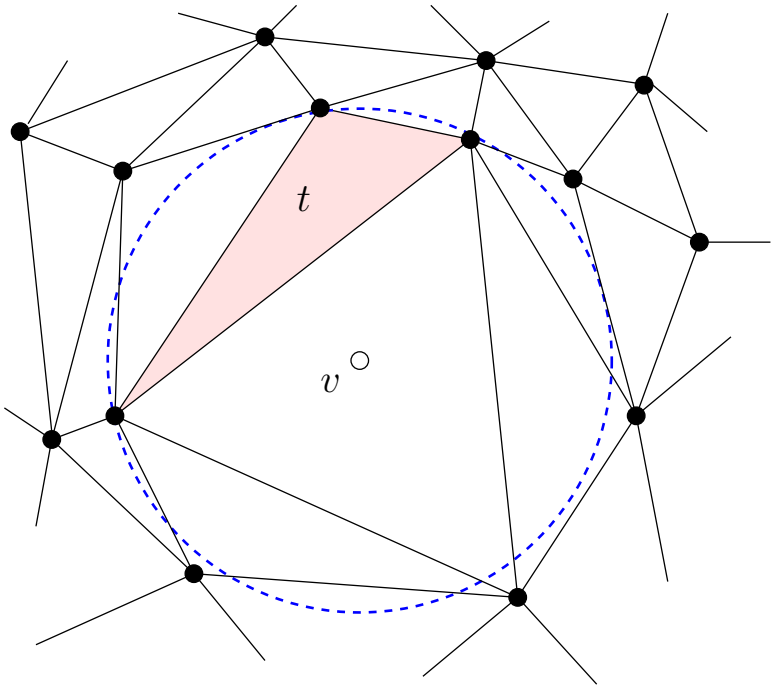
Quality of triangles

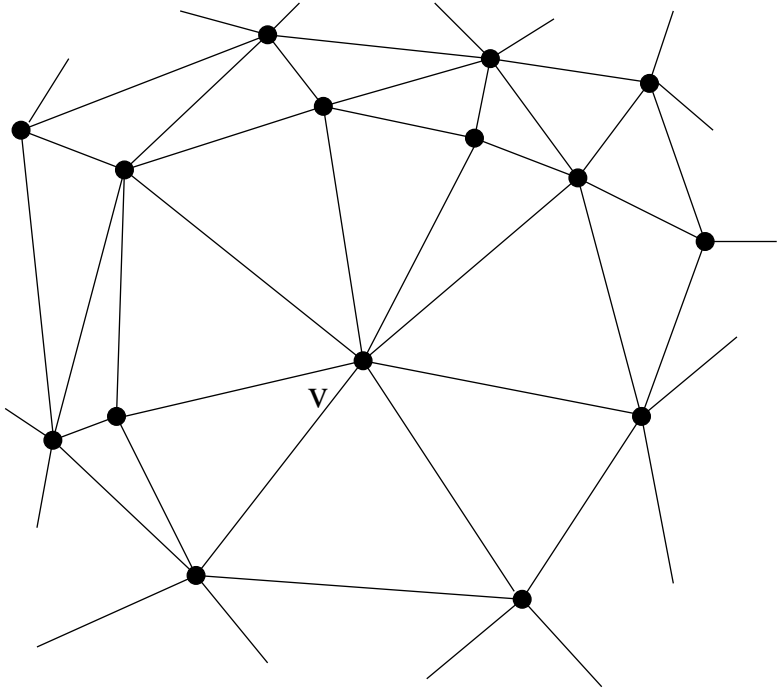
- minimal angle $\min_{\theta \text{ of } \tau} \theta$
- mean ratio $\sum_k d_k^2 / |\tau|^{2/n}$ (or its reciprocal)
- aspect/radius ratio *inradius*/*circumeradius*

There are lots of geometric qualities ... .

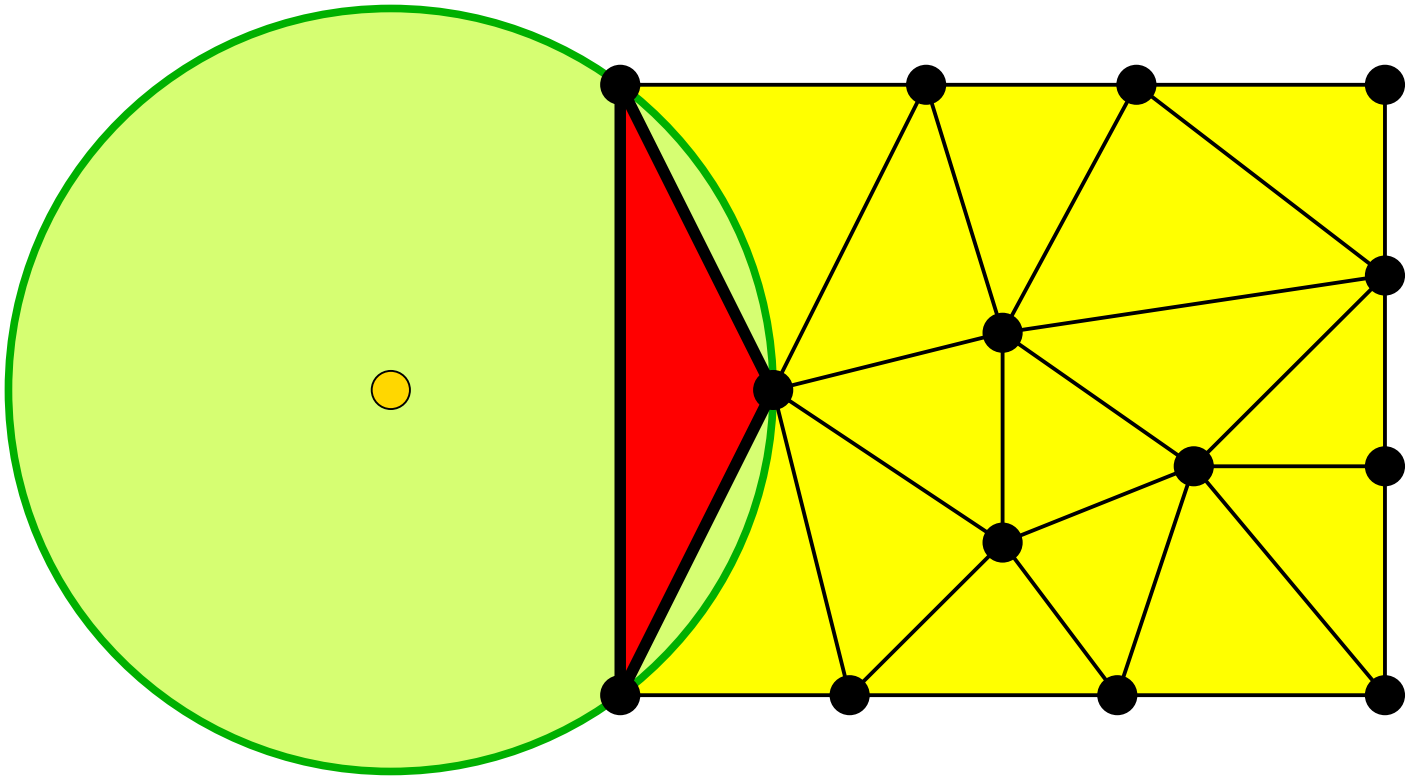
Delaunay refinement [Chew 1989, Ruppert 1995]

- Kill bad elements by insertion of their circumcenter.
- Bad elements: badly-shaped, oversized, etc.

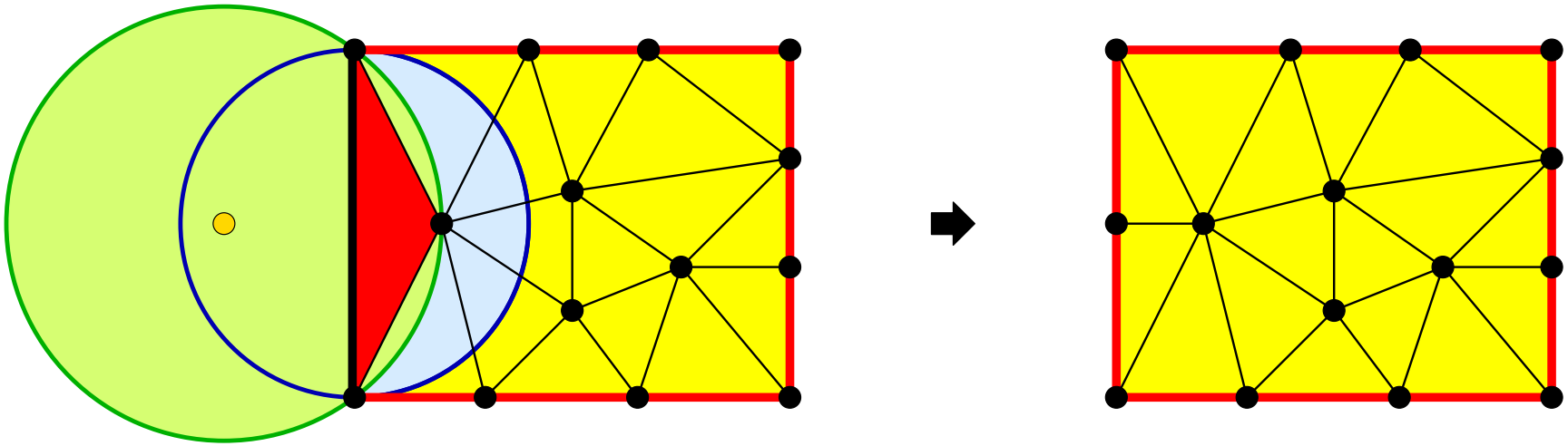




Circumcenter may lie outside of the domain



Boundary protection



Split segments if its diametral circumcircle is not empty

A result of Delaunay refinement